# Generative Structure from Motion with Native 3D Diffusion

**Haobin Duan**[1,2*] **Binbin Huang**[1,2*] **Mochu Xiang**[1,2] **Yiqun Zhao**[1,2]
**Zibo Zhao**[3] **Yi Ma**[1,2,4] **Shenghua Gao**[1,2,4†]

[1]The University of Hong Kong [2]Transengram [3]Tencent
[4]Shenzhen Loop Area Institute

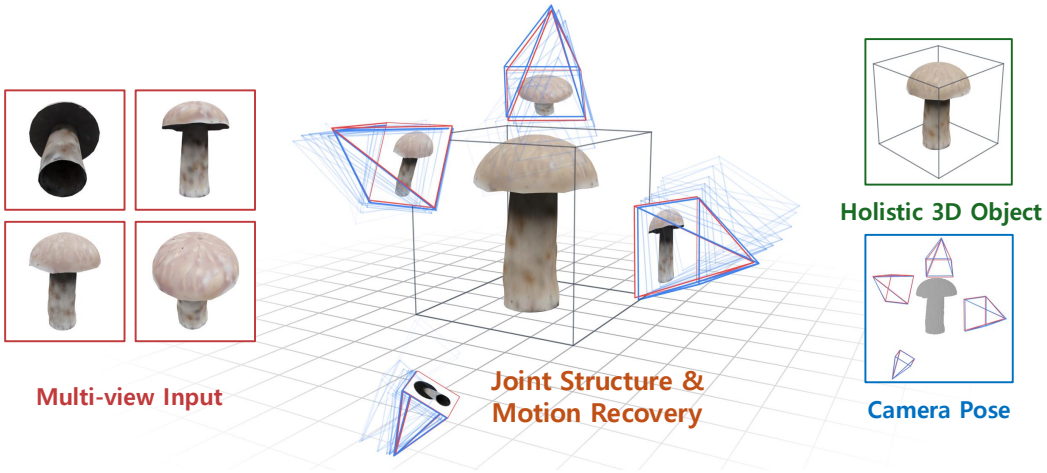


Figure 1: **Generative object shape and camera motion recovery.** Given multiview input images, **GENSFM** recovers the object shape and camera poses in a unified 3D space. A diffusion transformer performs multi-view reasoning through iterative structure-and-pose denoising (the trajectory of the camera), emulating an end-to-end SfM process. The denoised cameras (blue) align accurately with the ground-truth poses (red) and the recovered 3D structure gives a holistic object representation.

## Abstract

Recovering an object's 3D structure and camera motion from a few uncalibrated views is a fundamental problem in computer vision. Classical structure from motion methods recover 3D geometry consistent with the input images but cannot recover unseen regions; native generation methods produce plausible 3D objects which are related but not necessarily faithful to the input images. Methods simply concatenating reconstruction and generation allow error to cascade across the two stages, instead of producing a 3D object consistent with both the distribution and the input. We propose **GENSFM**, which integrates image-based 3D reconstruction with a native 3D diffusion model. In particular, we use a multi-modal diffusion transformer to jointly denoise the shape latent and per-view pose latents under shared self-attention, grounding generated geometry in the observed views while using object-level priors to infer unobserved regions. A compact 3D-native UV-volume tokenization keeps the joint sequence small as views are added, and a pose-aware stage refines geometry and texture by sampling per-voxel image features at the recovered camera poses. On Toys4k and GSO, for 4 uncalibrated

*Equal contribution.
†Corresponding author.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

views, **GENSFM** improves novel-view PSNR by 3.2–4.5 dB and reduces median Chamfer distance by 2.8–3.8× over a strong native-3D generation baseline, while recovering input-view poses over 96% Acc@5° without any external pose-estimation backbone, post-hoc optimization, or known intrinsics. Project page: https://dotb1ue.github.io/GenSfm/.

## 1 Introduction

Recovering 3D structure and camera motion from images is a foundational problem in computer vision. Classical Structure-from-Motion and Multi-View Stereo pipelines [1, 2, 3] estimate camera poses from multi-view correspondences and reconstruct geometry through photometric or geometric consistency, often regularized by hand-crafted priors. While effective when input views provide sufficient texture, overlap, and coverage, this geometric paradigm becomes brittle in sparse object-centric settings, where only a few uncalibrated views are available and large object regions go unrecovered.

Such settings are common in practice, from robotic manipulation to AR capture and content-creation pipelines, all of which need a holistic 3D representation to reason about occluded surfaces and contact regions. Sparse-view 3D object recovery therefore requires more than triangulating visible evidence: it must infer a complete 3D explanation that is both consistent with the input images and plausible as an object, jointly answering ***what*** the object is and ***where*** each view was taken from.

Recent learning-based methods relax the texture and overlap assumptions of classical pipelines by introducing large-scale data-driven priors. Feed-forward reconstructors [4, 5, 6] predict camera parameters and pixel-aligned 3D representations jointly, while pose-centric works [7, 8, 9, 10] estimate relative camera motion under limited overlap and pose ambiguity. Both threads remain primarily observation-driven: they recover what the input views support but do not explicitly model the distribution of complete object shapes, leaving outputs incomplete in unobserved regions when input coverage is poor.

A complementary line of work introduces generative priors learned from large-scale 3D datasets [11] into sparse reconstruction. View-based generative SfM [12, 13, 14, 15] casts structure and motion as a generative problem in view space, while reconstruct-then-generate pipelines [16, 17, 18, 19, 20] pair feed-forward reconstructors with native 3D generators [21, 22, 23]. The latter family in particular has shown that 3D generative priors can complete hidden geometry that observation-driven methods cannot. These systems remain cascaded: an external module predicts poses or image-aligned representations and the 3D generator only completes the result, propagating errors across stages and preventing joint optimization of a single 3D hypothesis that explains all input views.

In this paper, we invert this pipeline: instead of using a generative prior to complete the output of a feed-forward reconstructor, we treat the 3D generative prior itself as the primary reasoning mechanism for sparse-view reconstruction. We propose **GENSFM**, an end-to-end framework that formulates joint recovery of complete object structure and per-view camera motion as a single conditional inference problem in a native 3D diffusion model, as shown in Fig. 1. The diffusion prior operates directly on a native 3D representation throughout inference, simultaneously denoising shape and per-view pose under joint self-attention. This grounds the generated object in the observed views while using object-level regularities to infer unobserved geometry.

Realizing this formulation requires a single diffusion transformer that jointly denoises a shared 3D shape with a per-view pose latent bound to its input image, while keeping the per-view pose-token overhead small as views are added. We adopt CUPID's [24] 3D-native UV-volume pose tokenization, but compress each view's pose latent to a compact token grid, since the dense volume over-parameterizes a low-DoF perspective projection. A learned view embedding tags each view's pose latent and image tokens, binding pose to its source image. Since neither stream natively carries cross-view information, we prepend a small set of register tokens to the shape stream as a shared channel for all views to communicate through self-attention.

Instead of relying on back-projection to recover surface texture like feed-forward methods [4], we add a second stage of native 3D refinement and texturing [22] so that hidden regions stay coherent with the observed views. We design a pose-aware fine stage that, using the cameras predicted in the

Table 1: Comparison of paradigms for sparse-view object structure-and-motion recovery. "Where" denotes camera-motion recovery and "what" denotes object-structure recovery, with observed and hidden geometry referring to visible and unobserved object regions, respectively. "Unified joint inference" means jointly recovering complete object structure and camera motion in one model, while "native 3D prior" denotes a generative prior over complete 3D shapes. ✓: explicit support; △: partial or indirect support; –: not a primary objective.

| **Method family** | **Where:** | **What: object structure** | | **Unified joint** | **Native** |
|---|---|---|---|---|---|
| | **camera motion** | **observed** | **hidden** | **inference** | **3D prior** |
| Classical SfM/MVS [1, 2, 3] | ✓ | ✓ | – | △ | – |
| Pose-centric methods [7, 8, 9, 10] | ✓ | – | – | – | – |
| Feed-forward reconstruction [4, 5, 6] | ✓ | ✓ | △ | △ | – |
| Native 3D generation [21, 22, 23] | – | △ | ✓ | – | ✓ |
| View-based generative SfM [12, 13, 14] | ✓ | △ | △ | △ | – |
| Recon. with 3D Gen. priors [18, 17, 19] | ✓ | ✓ | ✓ | △ | ✓ |
| **GENSFM** (Ours) | ✓ | ✓ | ✓ | ✓ | ✓ |

coarse stage, projects every active voxel into the input views and conditions denoising on per-voxel semantic and texture features sampled and averaged across views.

On the Toys4k [25] and GSO [26] sparse-view object benchmarks under uncalibrated 4-view inputs, **GENSFM** improves novel-view PSNR by 3.2–4.5 dB, reduces median Chamfer distance by 2.8–3.8×, and lifts F@.01 from 0.56 to 0.83 on Toys4k over the strong native-3D generation baseline TRELLIS-M [22], while recovering input-view camera poses to over 96% Acc@5° (a 5× rotation-error reduction over the strongest pose baseline VGGT [6]) without requiring known intrinsics. Our contributions are summarized as follows:

- We propose **GENSFM**, a joint native-3D architecture that recasts a diffusion model from a post-hoc completion module into the primary inference mechanism for joint sparse-view structure and motion recovery, eliminating any external feed-forward reconstruction backbone.
- A 3D-native multi-view pose tokenization that places per-view UV-volume pose latents and the shape latent in a single multi-modal sequence, jointly denoised by a diffusion transformer under shared self-attention.
- A pose-aware fine stage that, with cameras recovered from the coarse stage, anchors semantic and texture features at every active voxel and averages them across views, sharpening surface detail beyond global, pose-agnostic conditioning.

## 2 Related Work

Sparse-view object reconstruction must jointly answer two coupled questions: ***what*** is the 3D object, and ***where*** was each view taken from? ***What*** requires both visible and hidden object structure, while ***where*** requires the camera motion relating the input views. Existing methods typically excel on one side or address the two with separate models stacked in series (Tab. 1).

**Feed-forward pose and 3D reconstruction.** Pose-centric approaches [7, 8, 9, 10] target the *where* side, regressing relative camera motion under limited overlap and pose ambiguity, but leave *what* largely untouched. More general feed-forward reconstructors [4, 5, 6] predict dense pointmaps, correspondences, and camera parameters jointly, addressing camera motion and visible geometry; hidden geometry, however, is recovered only indirectly through learned priors.

**Native 3D generation.** Native 3D generators trained on large 3D corpora [11], such as TripoSR [21], TRELLIS [22], and Hunyuan3D [23], are strong on *what*: complete object structure is modeled explicitly and hidden geometry is hallucinated from the prior. They do not target *where*: camera motion is typically assumed and only weakly encoded through cross-attention, so applied directly to uncalibrated sparse views they can produce plausible but misaligned geometry.

**View-based generative SfM.** View-based generative SfM [12, 13, 14, 15] casts *what* and *where* as a joint generative problem in view space; DiffusionSfM, for instance, diffuses ray origins and

endpoints to recover per-view geometry and camera motion jointly. Because diffusion operates over view-based quantities (rays, pointmaps, depths), hidden *what* is recovered only insofar as it can be back-projected from individual views.

**Reconstruction with 3D generative priors.** Reconstruct-then-generate pipelines [16, 17, 18, 19, 20, 27] couple feed-forward reconstruction with native 3D generators: e.g., ReconViaGen [18] injects VGGT [6] geometry tokens into a 3D generator. These pipelines recover hidden geometry through the generative prior while grounding camera motion and visible structure with feed-forward geometry, but solve *what* and *where* in separate stages, so errors from the reconstruction backbone propagate into a 3D hypothesis that need not remain coherent with all input views.

## 3 Preliminaries

**Overview.** Given $N$ uncalibrated multi-view images $\mathcal{I} = \{\mathbf{I}_i\}_{i=1}^N$ of an object $\mathcal{O}$, our goal is to recover the holistic 3D structure and per-view camera poses $\mathcal{P} = \{\mathbf{P}_i\}_{i=1}^N$ as a coarse-to-fine generative problem in a native 3D diffusion model. We build on two native 3D generators reviewed below: TRELLIS [22] provides the two-stage backbone and CFM objective we inherit, and CUPID [24] contributes the 3D-native single-image pose tokenization we lift to multi-view.

**Coarse-to-fine 3D generation.** Our backbone is TRELLIS [22], which generates a 3D object in two coarse-to-fine stages under a shared conditional flow matching [28] (CFM) objective. The first stage produces a sparse-structure (SS) latent $\mathbf{z}^s$ on a $16^3$ grid, decoded to a $64^3$ occupancy grid whose active voxels define positions $\{\mathbf{x}_j\}_{j=1}^L$. The second stage produces the structured-latent (SLat) set $\mathcal{S} = \{\mathbf{x}_j, \mathbf{f}_j\}_{j=1}^L$ with per-voxel features $\mathbf{f}_j$, which separate decoders [29, 30] map to 3D Gaussians, a radiance field, or a mesh. Both stages train with

$$\mathcal{L}_{\text{CFM}}(\phi) = \mathbb{E}_{t,\mathbf{z}_0,\boldsymbol{\epsilon}} \left\| \mathbf{v}_\phi(\mathbf{z}_t, \mathbf{I}, t) - (\boldsymbol{\epsilon} - \mathbf{z}_0) \right\|_2^2, \tag{1}$$

where $\mathbf{z}_t = (1-t)\mathbf{z}_0 + t\,\boldsymbol{\epsilon}$ is the noisy latent at $t \in [0,1]$ for clean target $\mathbf{z}_0$ ($\mathbf{z}^s$ for SS, $\{\mathbf{f}_j\}$ for SLat) and noise $\boldsymbol{\epsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$; $\mathbf{v}_\phi$ is the DiT velocity prediction conditioned on $\mathbf{I}$.

**3D-native pose representation.** Our pose tokenization derives from CUPID [24], which extends TRELLIS's SS stage to jointly recover the camera pose for a single input image: the SS latent gains channels encoding per-voxel UV coordinates, decoded to a $64^3$ UV volume that gives each active voxel's projection $\mathbf{u}_j \in [0,1]^2$. The $3\times4$ projection matrix $\mathbf{P}$ is thus a dense 3D-2D field on the active voxels, recovered from the per-voxel correspondences $\{\mathbf{x}_j, \mathbf{u}_j\}_{j=1}^L$ by Direct Linear Transform (DLT) least-squares [31]:

$$\mathbf{p}^* = \underset{\|\mathbf{p}\|_2=1}{\arg\min} \|A\mathbf{p}\|_2^2, \qquad \mathbf{P}^* = \text{reshape}_{3\times4}(\mathbf{p}^*), \tag{2}$$

where $A$ stacks the homogeneous 3D–2D correspondence equations and $\mathbf{p}$ vectorizes the projection matrix. A single SVD solves this linear system, followed by RQ decomposition to recover $\mathbf{K}$ and $[\mathbf{R} \mid \mathbf{t}]$, without iterative per-instance refinement. We use $\pi$ below for perspective projection.

## 4 Method

We frame this joint recovery as a coarse-to-fine generative problem in the latent space of a native 3D diffusion model (Fig. 2). Building on TRELLIS's two-stage architecture and CUPID's 3D-native pose representation (Sec. 3), Sec. 4.1 jointly generates 3D structure and multi-view poses with a multi-modal DiT over per-view pose tokens, and Sec. 4.2 refines geometry and appearance via per-voxel pose-sampled image conditioning.

### 4.1 3D structure and pose recovery from multi-view images

The coarse stage casts joint recovery of 3D structure and per-view camera poses as a single denoising problem: each camera is encoded as a per-view 3D-native pose token, and shape and pose are denoised jointly by a multi-modal DiT with joint self-attention over both streams.

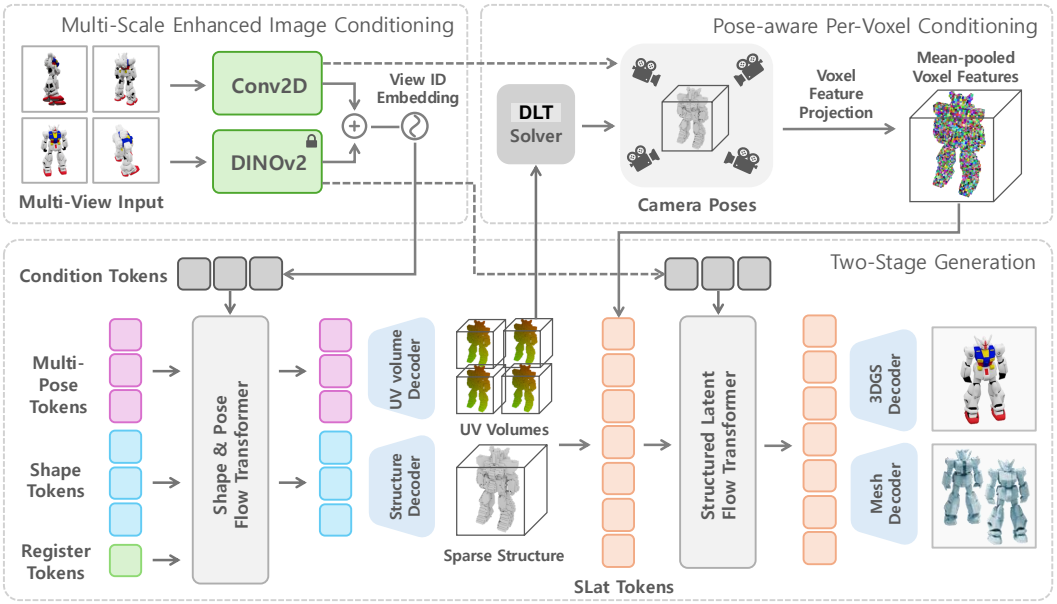


Figure 2: Overview of **GENSFM**. We cast pose-free multi-view 3D recovery as a single coarse-to-fine generative problem. The coarse stage (Sec. 4.1) jointly denoises the sparse-structure latent and per-view 3D-native pose tokens with a multi-modal shape-pose DiT, recovering shape and multi-view camera poses together. The fine stage (Sec. 4.2) refines geometry and appearance with per-voxel pose-sampled image conditioning, producing the final structured-latent set.

**Tokenizing multi-view camera poses.** For each view $i$, every active voxel $\mathbf{x}_j$ projects to $\mathbf{u}_{j,i} = \pi(\mathbf{P}_i, \mathbf{x}_j)$, yielding a UV volume $\mathbf{u}_i \in [0,1]^{2\times 64^3}$. A shared pose VAE compresses each $\mathbf{u}_i$ into a latent grid $\mathbf{z}_i^p \in \mathbb{R}^{c\times r\times r\times r}$ with $c$=8, $r$=4. To represent multiple input views, we form a unified token sequence with the shape latent $\mathbf{z}^s$ followed by the per-view pose latents $\{\mathbf{z}_i^p\}_{i=1}^N$ — in contrast to CUPID [24], whose channel-wise concatenation of shape and pose into a single SS latent is restricted to one view. We choose $r$=4 (vs CUPID's $r$=16), giving 64× fewer tokens per pose at negligible accuracy cost since the UV volume is heavily over-parameterized for a low-DoF projection. At inference, the pose VAE decodes each generated $\mathbf{z}_i^p$ back to a UV volume, from which $\mathbf{P}_i^*$ is recovered by solving Eq. 2. See App. A.3 for details.

**Multi-view conditioned shape-pose DiT.** A shared transformer backbone denoises shape and pose jointly, with stream-specific projections and grid-aligned positional embeddings. Learnable register tokens $\mathbf{z}^r$ are prepended to the shape stream to capture global context. We extend the CFM interpolation of Eq. 1 to a shape timestep $t^s \in [0,1]$ and per-view pose timesteps $\mathbf{t}^p \in [0,1]^N$, giving $\mathbf{z}_0 = [\mathbf{z}^s; \mathbf{z}^p]$ and $\mathbf{z}_t = [\mathbf{z}_t^s; \mathbf{z}_t^p]$. Stream-specific in-projections then lift each noisy latent to the shared transformer width, where positional and view embeddings are added in to form the layer-0 hidden state of the DiT:

$$\mathbf{h}_0 = [\,\mathbf{z}^r;\;\; W_{\text{in}}^s\,\mathbf{z}_t^s + \mathbf{e}^s;\;\; W_{\text{in}}^p\,\mathbf{z}_t^p + \mathbf{e}^p + \mathbf{e}^v\,], \tag{3}$$

where $\mathbf{e}^s, \mathbf{e}^p$ are absolute 3D PEs over the $16^3$ and $4^3$ grids that anchor each stream to its native voxel layout, and $\mathbf{e}^v$ is a per-view ID embedding shared between each pose latent and its source view's image-condition tokens (Eq. 6), binding each pose to its view through cross-attention. We then stack $L$ DiT blocks; each block injects the multi-view condition tokens $\mathbf{z}^I = \{\mathbf{z}_i^I\}_{i=1}^N$,

$$\mathbf{h}_{l+1} = \text{CrossAttn}\big(Q(\mathbf{h}_l),\, K(\mathbf{z}^I),\, V(\mathbf{z}^I)\big), \tag{4}$$

where $Q, K, V$ are the query/key/value projections, applied jointly over all $N$ views without per-view masking. Each block also runs a joint self-attention over the entire sequence and a feed-forward sub-layer, both modulated by per-modality AdaLN values conditioned on $t^s$ for shape tokens and on $t_i^p$ for view-$i$ pose tokens. After the final block, stream-specific out-projections read out the velocities (register outputs are discarded), $\mathbf{v}_\phi(\mathbf{z}_t, \mathbf{z}^I, t^s, \mathbf{t}^p) = [\,W_{\text{out}}^s\,\mathbf{h}_L^s;\; W_{\text{out}}^p\,\mathbf{h}_L^p\,]$, and the whole network is trained with a stream-wise conditional flow matching loss:

$$\mathcal{L}_{\text{coarse}}(\phi) = \mathbb{E}_{t^s,\,\mathbf{t}^p,\,\mathbf{z}_0,\,\boldsymbol{\epsilon}}\left\|\mathbf{v}_\phi(\mathbf{z}_t, \mathbf{z}^I, t^s, \mathbf{t}^p) - (\boldsymbol{\epsilon} - \mathbf{z}_0)\right\|_2^2. \tag{5}$$

Table 2: **Comparison with feed-forward reconstruction methods.** We align predictions to the ground truth via a similarity transform fit by least squares between predicted and ground-truth input-view pointmaps, then evaluate photometric quality on novel views and 3D geometric accuracy in the ground-truth coordinate frame. CD is bidirectional symmetric (squared L2, $\times 10^{-3}$), reported as the median across objects.

| Method | Toys4k | | | | | | GSO | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | CD-med↓ | F@.01↑ | F@.05↑ | PSNR↑ | SSIM↑ | LPIPS↓ | CD-med↓ | F@.01↑ | F@.05↑ |
| FreeSplatter [32] | 22.00 | 0.9183 | 0.0913 | 2.7160 | 0.3163 | 0.7112 | 20.07 | 0.8948 | 0.1135 | 3.2859 | 0.2713 | 0.6736 |
| LucidFusion [33] | 22.24 | 0.9251 | 0.0827 | 4.1596 | 0.1527 | 0.6670 | 20.94 | 0.9097 | 0.0943 | 3.8702 | 0.1755 | 0.7164 |
| ReconViaGen [18] | 21.88 | 0.9216 | 0.0721 | 1.1237 | 0.3369 | 0.8434 | 20.88 | 0.9021 | 0.0856 | 1.0495 | 0.3524 | 0.8789 |
| **GENSFM** | 28.61 | 0.9609 | 0.0308 | 0.2184 | 0.6701 | 0.9823 | 26.36 | 0.9411 | 0.0472 | 0.2801 | 0.6095 | 0.9842 |

**Multi-scale enhanced image condition.** DINOv2 patch features capture rich semantics but are spatially coarse, providing limited cues for the fine appearance and accurate 2D-3D correspondences our coarse stage must recover. We therefore augment them with multi-scale local features extracted by a small convolutional pyramid on the same image: for each view $i$, the condition tokens are

$$\mathbf{z}_i^I = \big(\text{DINOv2}(\mathbf{I}_i) \oplus \Phi_{\text{cnn}}^s(\mathbf{I}_i)\big) + \mathbf{e}^v, \tag{6}$$

where $\text{DINOv2}(\mathbf{I}_i)$ are the frozen patch tokens, $\Phi_{\text{cnn}}^s(\mathbf{I}_i)$ stacks features from a 4-level convolutional pyramid where each level patchified to the DINOv2 patch grid and channel-concatenated across levels, $\oplus$ denotes channel-wise concatenation per token, and $\mathbf{e}^v$ is the view-tag embedding from Eq. 3. The resulting $\mathbf{z}^I = \{\mathbf{z}_i^I\}_{i=1}^N$ serves as the cross-attention key/value sequence in every DiT block. See App. A.4 for details.

### 4.2 Pose-aware generation of fine geometry and appearance

The fine stage generates geometry and appearance with pose-aware per-voxel image conditioning: each voxel receives a semantic feature from DINOv2 and a texture feature from a multi-scale CNN pyramid, with both features average-pooled over pose-aligned projections across views.

**Pose-aware per-voxel image conditioning.** TRELLIS's SLat stage conditions on the input image only through global cross-attention with DINOv2 patch tokens, leaving fine-appearance cues unanchored from any specific voxel and producing blurry surface detail. Once $\mathbf{P}^*$ is recovered by the coarse stage (Sec. 4.1), every active voxel $\mathbf{x}_j$ has a known image projection $\mathbf{u}_j = \pi(\mathbf{P}^*, \mathbf{x}_j)$, so image features can be sampled directly per voxel. At each $\mathbf{x}_j$ we form two complementary features: a semantic feature $\mathbf{c}_j^{\text{sem}}$ from pose-sampled DINOv2 patches re-encoded through the frozen SLat encoder $\text{Enc}_\text{L}$ (placing it on the same latent manifold as the SLat target $\mathbf{f}_j$), and a texture feature $\mathbf{c}_j^{\text{tex}}$ from a multi-scale CNN pyramid $\Phi_{\text{cnn}}^l$ that supplies the low-level cues that DINOv2 lacks:

$$\begin{aligned} \{\mathbf{c}_j^{\text{sem}}\}_{j=1}^L &= \text{Enc}_\text{L}\left(\{\mathbf{x}_j, \text{BilinearInterp}(\mathbf{u}_j, \text{DINOv2}(\mathbf{I}))\}_{j=1}^L\right), \\ \mathbf{c}_j^{\text{tex}} &= \text{BilinearInterp}\left(\mathbf{u}_j, \Phi_{\text{cnn}}^l(\mathbf{I})\right). \end{aligned} \tag{7}$$

where BilinearInterp denotes bilinear interpolation. We channel-wise concatenate $\mathbf{c}_j^{\text{sem}}$ and $\mathbf{c}_j^{\text{tex}}$ with the noisy voxel feature $\mathbf{f}_j^t$ (the SLat target $\mathbf{f}_j$ noised via Eq. 1) and linearly project to form the per-voxel layer-0 DiT token $\mathbf{h}_0(\mathbf{x}_j) = W_{\text{in}}^l[\,\mathbf{f}_j^t \oplus \mathbf{c}_j^{\text{sem}} \oplus \mathbf{c}_j^{\text{tex}}\,]$; the remainder of the DiT follows TRELLIS. See App. A.4 for the visual pyramid architecture.

**Pooling per-voxel features across views.** With $N$ input views, every active voxel has $N$ pose-aligned projections $\mathbf{u}_{j,i} = \pi(\mathbf{P}_i^*, \mathbf{x}_j)$. We extend BilinearInterp to per-view sampling followed by averaging, $\text{BilinearInterp}(\mathbf{u}_j, \Phi(\mathcal{I})) := \frac{1}{N}\sum_{i=1}^N \text{BilinearInterp}(\mathbf{u}_{j,i}, \Phi(\mathbf{I}_i))$, so Eq. 7 applies unchanged with $\mathbf{I}$ replaced by $\mathcal{I}$. The cross-attention condition is likewise the per-view mean of DINOv2 patch tokens. The mean operator itself is parameter-free and view-count-agnostic; the model in this work is trained with $N$=4.

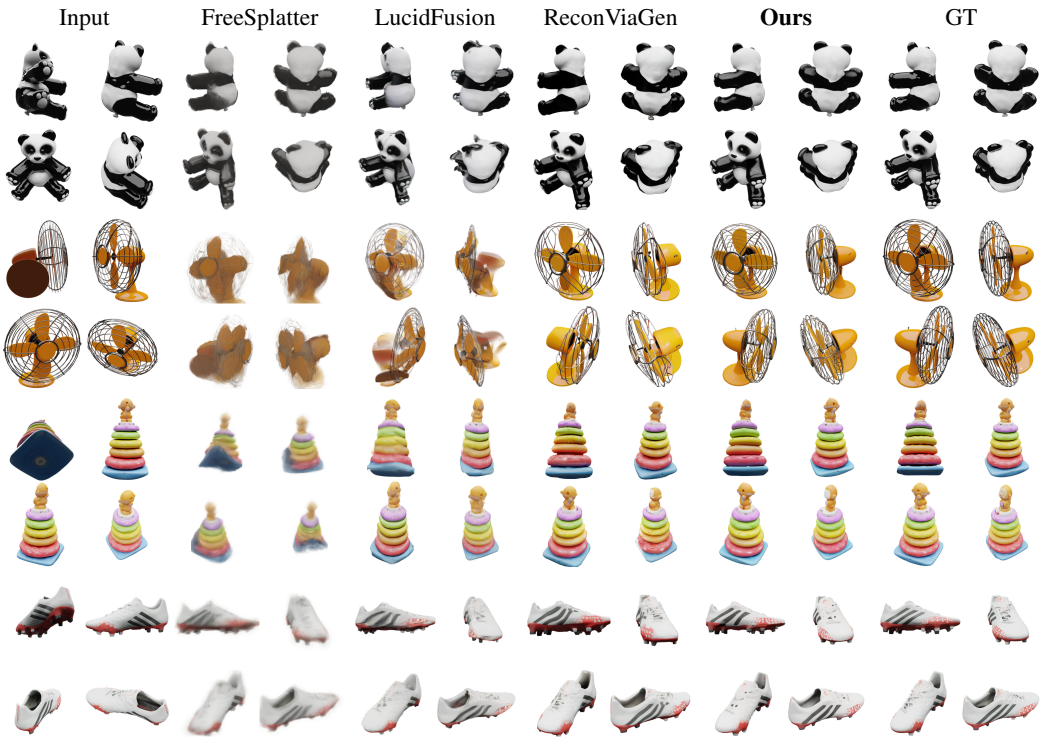


Figure 3: **Qualitative comparison with feed-forward reconstruction methods** on Toys4k (top two rows) and GSO (bottom two). All non-input columns render the same held-out novel view in the GT camera frame after the input-view pointmap LSQ alignment of Table 2.

## 5 Experiments

### 5.1 Experimental Setup

**Datasets.** We train our model on four public 3D-asset datasets: Objaverse-XL [11] (GitHub and Sketchfab partitions), HSSD [34], 3D-FUTURE [35], and ABO [36]. After TRELLIS's aesthetic filter ($\geq 4.5$), about 430K instances remain. Each is rendered following TRELLIS's protocol, with 24 condition views at $1024\times1024$ from cameras uniformly sampled on the upper hemisphere. We evaluate on two held-out test sets, Toys4k [25] (synthetic) and GSO [26] (real-captured), neither overlapping the training corpus.

**Evaluation protocol.** We assess photometric fidelity, 3D geometry, and camera-pose accuracy. Each test set uses its first 400 objects with two disjoint 4-view sets, $\{0, 6, 12, 18\}$ and $\{3, 9, 15, 21\}$, yielding 800 object–viewset instances for aggregation. As baselines fall into three capability families, we apply the most suitable evaluation protocol to each: (i) least squares on predicted vs ground-truth input-view pointmaps for feed-forward reconstruction, tight only with accurate predicted geometry and poses; (ii) similarity ICP on sampled point clouds for 3D generation; and (iii) gauge-invariant pair-wise relative-pose comparison on unordered input-view pairs for pose-predicting methods. **GenSfM** is reevaluated under each applicable protocol; comparisons are restricted to methods sharing a table and protocol. We report PSNR, SSIM, and LPIPS [37] on $512\times512$ novel views at the 8 held-out condition-view indices $\{2, 4, 8, 10, 14, 16, 20, 22\}$; bidirectional Chamfer Distance (CD; reported as the median across objects, since a few severe failures dominate the mean) and F-scores at $\tau \in \{0.01,\ 0.05\}$; and Relative Rotation Error (RRE) with Acc@$\{5°, 15°\}$, Relative Translation Error (RTE), and mean absolute FOV error (RFOV; only for methods that predict intrinsics). See App. A.1 for the full alignment protocol and App. A.2 for the pose metric definitions.

**Baseline methods.** We compare against a representative set of multi-view image-to-3D baselines: (a) **TRELLIS-S** [22]: TRELLIS run as a single-image generator on each input view independently,

Table 3: **Comparison with 3D generation methods.** For methods that produce a canonical 3D output without input camera poses, we align predictions to the ground truth via similarity ICP on 100K points sampled from the rendered surface of the output mesh of each side, then evaluate photometric quality on novel views and 3D geometric accuracy in the ground-truth coordinate frame.

| Method | Toys4k | | | | | | GSO | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | PSNR↑ | SSIM↑ | LPIPS↓ | CD-med↓ | F@.01↑ | F@.05↑ | PSNR↑ | SSIM↑ | LPIPS↓ | CD-med↓ | F@.01↑ | F@.05↑ |
| SAM 3D [19] | 20.49 | 0.9155 | 0.0926 | 1.7935 | 0.3292 | 0.8250 | 19.49 | 0.8995 | 0.1112 | 1.3716 | 0.3387 | 0.8505 |
| TRELLIS-S [22] | 21.78 | 0.9205 | 0.0874 | 1.2032 | 0.4064 | 0.8881 | 20.00 | 0.8943 | 0.1127 | 1.2418 | 0.3663 | 0.8695 |
| TRELLIS-M [22] | 24.01 | 0.9355 | 0.0658 | 0.4329 | 0.5570 | 0.9604 | 22.28 | 0.9138 | 0.0867 | 0.5200 | 0.5079 | 0.9655 |
| ReconViaGen [18] | 23.26 | 0.9316 | 0.0587 | 0.6955 | 0.4374 | 0.9555 | 21.98 | 0.9108 | 0.0778 | 0.6500 | 0.4428 | 0.9434 |
| **GENSFM** | 28.54 | 0.9599 | 0.0315 | 0.1133 | 0.8279 | 0.9970 | 25.50 | 0.9349 | 0.0533 | 0.1882 | 0.7283 | 0.9914 |




Figure 4: **Qualitative comparison with 3D generation methods** on Toys4k (top row) and GSO (bottom three). All non-input columns render the same held-out novel view in the GT camera frame after the similarity ICP alignment of Tab. 3.

with the mean over the four outputs reported; (b) **TRELLIS-M** [22]: TRELLIS with multi-view input in stochastic mode, where each denoising step is conditioned on one randomly chosen input view; (c) **SAM 3D** [19]: a single-image-to-3D generator run once per input view, with the mean over the four outputs reported; (d) **FreeSplatter** [32]: predicts per-view pixel-aligned 3D Gaussians together with input camera parameters in a pose-free setting; (e) **LucidFusion** [33]: predicts per-view relative coordinate maps that are fused into a 3D Gaussian output; (f) **ReconViaGen** [18]: combines feed-forward reconstruction with generative refinement, using VGGT-derived geometry tokens to condition a 3D generator; (g) **VGGT** [6]: a transformer that jointly regresses per-view pointmaps and camera extrinsics/intrinsics from uncalibrated images. Our table appear in three families: 3D generation (a,b,c,f), feed-forward reconstruction (d,e,f), and pose prediction (d,f,g).

### 5.2 Main Results

**Comparison with feed-forward reconstruction methods.** Tab. 2 and Fig. 3 compare against feed-forward reconstruction baselines under input-view pointmap LSQ alignment. **GENSFM** outperforms every baseline on every metric and on both datasets, with the gap widening at the strict F@.01 threshold and qualitatively in the form of alignment artifacts and surface noise. Feed-forward

Table 4: **Camera-pose prediction accuracy.** We report relative rotation error (RRE) with Acc.@$\{5°, 15°\}$ thresholds, relative translation error (RTE; the angular error between predicted and GT relative-translation directions), and mean absolute FOV error (RFOV; intrinsic-only); see supplementary for the full evaluation protocol. **GENSFM** reports two rows: known intrinsics (GT FOV fed in; RFOV not applicable, marked –) and predicted intrinsics. Angular pose metrics use raw, gauge-invariant pairwise relative poses and the same ambiguity-tolerant score for all methods.

| Method | Toys4k | | | | | GSO | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | RRE↓ | Acc.@5°↑ | Acc.@15°↑ | RTE↓ | RFOV↓ | RRE↓ | Acc.@5°↑ | Acc.@15°↑ | RTE↓ | RFOV↓ |
| FreeSplatter [32] | 26.45 | 0.2438 | 0.4996 | 32.32 | 12.51 | 29.10 | 0.2327 | 0.4646 | 32.81 | 13.13 |
| ReconViaGen [18] | 16.54 | 0.2044 | 0.6881 | 19.56 | 9.54 | 14.48 | 0.2260 | 0.7343 | 16.16 | 8.95 |
| VGGT [6] | 10.29 | 0.5969 | 0.8406 | 18.61 | 37.66 | 9.33 | 0.6310 | 0.8702 | 17.69 | 36.08 |
| **GENSFM** (predicted intrinsics) | 1.60 | 0.9723 | 0.9913 | 5.51 | 2.90 | 1.27 | 0.9821 | 0.9985 | 5.08 | 2.86 |
| **GENSFM** (known intrinsics) | 1.44 | 0.9798 | 0.9910 | 2.38 | – | 1.13 | 0.9892 | 0.9981 | 2.11 | – |

Table 5: **Ablation study on Toys4k.** Rows add components cumulatively, ending with the full model. Stage 1 uses a single-view-trained fine decoder extended to multi-view at inference; Stage 2 replaces it with a natively multi-view-trained decoder. All rows use the input-view pointmap alignment from Table 2; RRE and Acc.@5° use a similarity fit from the final pointmap, so the fine decoder can shift them without changing raw coarse cameras. CD is the median across objects.

| Stage | Variant | PSNR↑ | LPIPS↓ | CD-med↓ | F@.01↑ | F@.05↑ | RRE↓ | Acc.@5°↑ |
|---|---|---|---|---|---|---|---|---|
| Stage 1 | Base (per-view cross-attention) | 27.00 | 0.0383 | 0.2811 | 0.6078 | 0.9677 | 5.62 | 0.8653 |
| | + full cross-attention | 27.49 | 0.0359 | 0.2568 | 0.6324 | 0.9748 | 5.20 | 0.9069 |
| | + visual conditioning | 27.65 | 0.0353 | 0.2488 | 0.6346 | 0.9783 | 4.59 | 0.9228 |
| | + register tokens | 28.23 | 0.0327 | 0.2204 | 0.6670 | 0.9824 | 3.55 | 0.9478 |
| Stage 2 | + native multi-view training (**Full model**) | 28.61 | 0.0308 | 0.2184 | 0.6701 | 0.9823 | 3.65 | 0.9416 |

methods compound any disagreement between their predicted geometry and predicted poses, while a generative 3D-native model refines it away.

**Comparison with 3D generation methods.** Tab. 3 and Fig. 4 compare against 3D generation models that produce a canonical, pose-free output, under similarity-ICP alignment. **GENSFM** leads on every metric, with the largest margins on the strict-threshold geometry metrics (F@.01, CD-med). Pose-free baselines hallucinate plausible but mismatched detail on unseen sides; the two strong multi-view baselines (TRELLIS-M's stochastic per-step view conditioning and ReconViaGen's feed-forward reconstruction with generative refinement) still trail **GENSFM** by large margins, indicating the gain comes from joint shape–pose denoising rather than late-stage view sampling.

**Pose prediction accuracy.** Tab. 4 reports camera-pose accuracy on baselines that predict input-view extrinsics. **GENSFM** achieves substantially lower rotation and translation errors, with Acc.@5° saturating; baselines commonly fail on symmetric objects whose orientation depends on low-level visual cues. Our model also recovers intrinsics accurately (known-intrinsics row shown for reference); VGGT, not trained on varying-FOV synthetic data, incurs large RFOV despite strong extrinsics, whereas **GENSFM** keeps RFOV under 3° on both datasets while saturating Acc.@5°. Thus, RFOV confirms reliable intrinsic recovery under varying FOV, but its margin over VGGT also reflects training-distribution coverage, not architecture alone.

### 5.3 Ablation Study

We ablate each design choice on Toys4k in Tab. 5, with rows added cumulatively. In Stage 1, full cross-attention enables cross-view information sharing and improves all metrics; visual feature conditioning sharpens photometric fidelity and pose accuracy with only marginal geometry gains, suggesting appearance cues primarily help rendering and pose grounding; shared register tokens aggregate global information and contribute most to pose accuracy and the strict F@.01 threshold. In Stage 2, a fine decoder trained natively on multi-view data sharpens photometric fidelity, while 3D geometry is already saturated. The decoder swap leaves raw coarse-stage cameras unchanged but shifts geometry-aligned pose metrics because their similarity fit depends on the final pointmap. Fig. 5 corroborates these trends qualitatively: surface details (e.g., the screwdriver cap, the snake's head) sharpen left-to-right and the full model best matches GT.

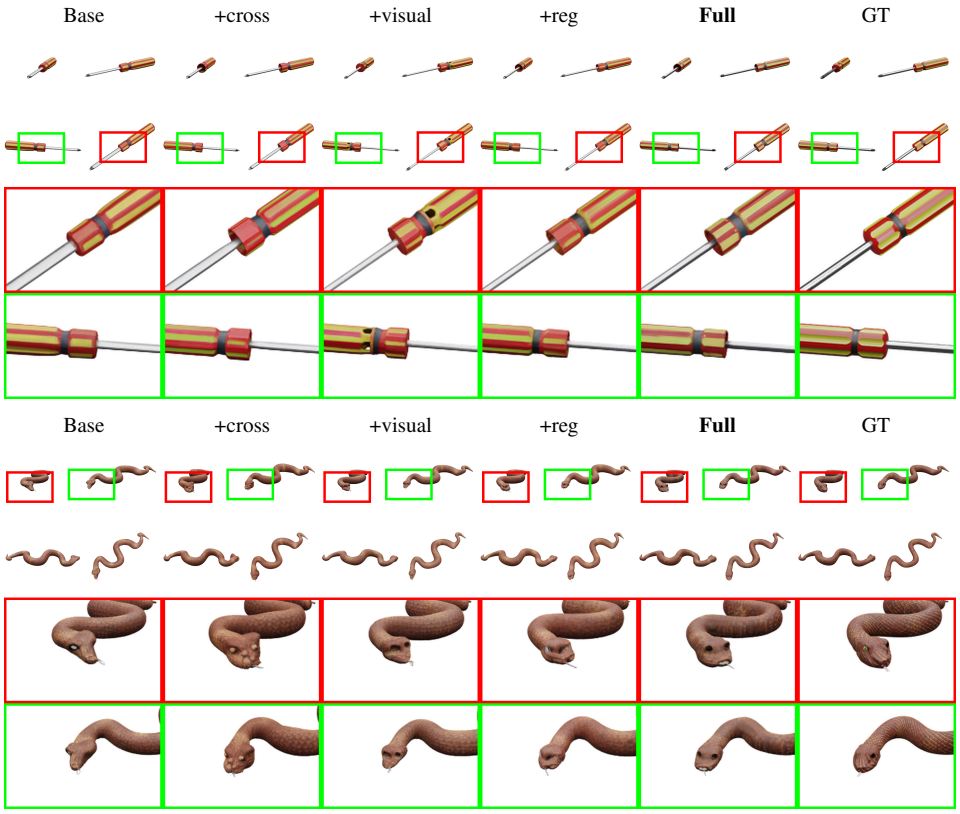


Figure 5: **Qualitative ablation on Toys4k.** For each instance we show the four held-out test views as a $2\times2$ grid (top), with two zoomed-in regions of largest cross-variant disagreement below (red / green borders). Columns left-to-right correspond to the cumulative ablation rows of Table 5: Base, +cross, +visual, +reg, the full model, and ground truth. Each added component sharpens local geometry and texture, with the full model best matching GT.

## 6 Conclusion

We presented **GENSFM**, which recasts a native 3D diffusion model as the primary inference mechanism for sparse-view object reconstruction, jointly denoising shape and per-view camera poses in a single multi-modal diffusion transformer. Three choices keep the formulation tractable: a UV-volume pose latent compressed to a compact token grid, a unified multi-modal sequence that places shape and per-view pose tokens in a single self-attention, and a pose-aware fine stage that conditions each voxel on features sampled at its pose-aligned image projection. On Toys4k and GSO, **GENSFM** outperforms feed-forward, 3D-generation, and pose-prediction baselines on photometric, geometric, and pose metrics simultaneously, recovering both extrinsics and intrinsics to high accuracy without any external pose estimator or post-hoc optimization.

**Limitations and future work.** Our results are restricted to object-centric inputs at $N=4$; extending to larger view counts, scene-level inputs, and in-the-wild captures with looser segmentation is a natural next step. Under distribution shift, large pose/calibration errors may misregister fine-stage image features, degrading local refinement while the coarse 3D prior preserves global completion. The model also inherits the object distribution and resolution limits of its TRELLIS backbone, so scaling the underlying generator should directly improve sparse-view recovery.

## Acknowledgments and Disclosures

This work is partially supported by funding and computing resources provided by the HKU-Shanghai Intelligent Computing Research Center. This work was supported by the General Research Fund of the Research Grants Council of Hong Kong under Grant No. 17200725 and the JC STEM Lab funded by The Hong Kong Jockey Club Charities Trust. The authors declare no competing interests.

## References


[1] Johannes Lutz Schönberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2, 3

[2] Johannes Lutz Schönberger, Enliang Zheng, Marc Pollefeys, and Jan-Michael Frahm. Pixelwise view selection for unstructured multi-view stereo. In *European Conference on Computer Vision (ECCV)*, 2016. 2, 3

[3] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009. 2, 3

[4] Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. Dust3r: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 20697–20709, 2024. 2, 3

[5] Vincent Leroy, Yohann Cabon, and Jérôme Revaud. Grounding image matching in 3d with mast3r. In *European conference on computer vision*, pages 71–91. Springer, 2024. 2, 3

[6] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 5294–5306, 2025. 2, 3, 4, 8, 9

[7] Jason Y Zhang, Deva Ramanan, and Shubham Tulsiani. Relpose: Predicting probabilistic relative rotation for single objects in the wild. In *European Conference on Computer Vision*, pages 592–611. Springer, 2022. 2, 3

[8] Jianyuan Wang, Christian Rupprecht, and David Novotny. Posediffusion: Solving pose estimation via diffusion-aided bundle adjustment. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9773–9783, 2023. 2, 3

[9] Samarth Sinha, Jason Y Zhang, Andrea Tagliasacchi, Igor Gilitschenski, and David B Lindell. Sparsepose: Sparse-view camera pose regression and refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 21349–21359, 2023. 2, 3

[10] Jason Y Zhang, Amy Lin, Moneish Kumar, Tzu-Hsuan Yang, Deva Ramanan, and Shubham Tulsiani. Cameras as rays: Pose estimation via ray diffusion. *arXiv preprint arXiv:2402.14817*, 2024. 2, 3

[11] Matt Deitke, Ruoshi Liu, Matthew Wallingford, Huong Ngo, Oscar Michel, Aditya Kusupati, Alan Fan, Christian Laforte, Vikram Voleti, Samir Yitzhak Gadre, et al. Objaverse-xl: A universe of 10m+ 3d objects. *Advances in Neural Information Processing Systems*, 36:35799–35813, 2023. 2, 3, 7

[12] Qitao Zhao, Amy Lin, Jeff Tan, Jason Y. Zhang, Deva Ramanan, and Shubham Tulsiani. Diffusionsfm: Predicting structure and motion via ray origin and endpoint diffusion. In *CVPR*, 2025. 2, 3

[13] Qihang Zhang, Shuangfei Zhai, Miguel Ángel Bautista, Kevin Miao, Alexander Toshev, Joshua Susskind, and Jiatao Gu. World-consistent video diffusion with explicit 3d modeling. In *arXiv*, 2024. 2, 3

[14] Chao Xu, Ang Li, Linghao Chen, Yulin Liu, Ruoxi Shi, Hao Su, and Minghua Liu. Sparp: Fast 3d object reconstruction and pose estimation from sparse views. In *European Conference on Computer Vision*, pages 143–163. Springer, 2024. 2, 3

[15] Qitao Zhao and Shubham Tulsiani. Sparse-view pose estimation and reconstruction via analysis by generative synthesis. In *NeurIPS*, 2024. 2, 3

[16] Andreea Ardelean, Mert Özer, and Bernhard Egger. Gen3dsr: Generalizable 3d scene reconstruction via divide and conquer from a single view. In *2025 International Conference on 3D Vision (3DV)*, pages 616–626. IEEE, 2025. 2, 4

[17] Zhisheng Huang, Jiahao Chen, Cheng Lin, Chenyu Hu, Hanzhuo Huang, Zhengming Yu, Mengfei Li, Yuheng Liu, Zekai Gu, Zibo Zhao, et al. Unirecgen: Unifying multi-view 3d reconstruction and generation. *arXiv preprint arXiv:2604.01479*, 2026. 2, 3, 4

[18] Jiahao Chang, Chongjie Ye, Yushuang Wu, Yuantao Chen, Yidan Zhang, Zhongjin Luo, Chenghong Li, Yihao Zhi, and Xiaoguang Han. Reconviagen: Towards accurate multi-view 3d object reconstruction via generation. *arXiv preprint arXiv:2510.23306*, 2025. 2, 3, 4, 6, 8, 9

[19] SAM 3D Team, Xingyu Chen, Fu-Jen Chu, Pierre Gleize, Kevin J Liang, Alexander Sax, Hao Tang, Weiyao Wang, Michelle Guo, Thibaut Hardin, Xiang Li, Aohan Lin, Jiawei Liu, Ziqi Ma, Anushka Sagar, Bowen Song, Xiaodong Wang, Jianing Yang, Bowen Zhang, Piotr Dollár, Georgia Gkioxari, Matt Feiszli, and Jitendra Malik. Sam 3d: 3dfy anything in images. 2025. 2, 3, 4, 8

[20] Yawar Siddiqui, Duncan Frost, Samir Aroudj, Armen Avetisyan, Henry Howard-Jenkins, Daniel DeTone, Pierre Moulon, Qirui Wu, Zhengqin Li, Julian Straub, et al. Shaper: Robust conditional 3d shape generation from casual captures. *arXiv preprint arXiv:2601.11514*, 2026. 2, 4

[21] Dmitry Tochilkin, David Pankratz, Zexiang Liu, Zixuan Huang, Adam Letts, Yangguang Li, Ding Liang, Christian Laforte, Varun Jampani, and Yan-Pei Cao. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*, 2024. 2, 3

[22] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 21469–21480, 2025. 2, 3, 4, 7, 8

[23] Zibo Zhao, Zeqiang Lai, Qingxiang Lin, Yunfei Zhao, Haolin Liu, Shuhui Yang, Yifei Feng, Mingxin Yang, Sheng Zhang, Xianghui Yang, et al. Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation. *arXiv preprint arXiv:2501.12202*, 2025. 2, 3

[24] Binbin Huang, Haobin Duan, Yiqun Zhao, Zibo Zhao, Yi Ma, and Shenghua Gao. Cupid: Generative 3d reconstruction via joint object and pose modeling. *arXiv preprint arXiv:2510.20776*, 2025. 2, 4, 5

[25] Stefan Stojanov, Anh Thai, and James M Rehg. Using shape to categorize: Low-shot learning with an explicit shape bias. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1798–1808, 2021. 3, 7

[26] Laura Downs, Anthony Francis, Nate Koenig, Brandon Kinman, Ryan Hickman, Krista Reymann, Thomas B McHugh, and Vincent Vanhoucke. Google scanned objects: A high-quality dataset of 3d scanned household items. In *2022 International Conference on Robotics and Automation (ICRA)*, pages 2553–2560. IEEE, 2022. 3, 7

[27] Kaixin Yao, Longwen Zhang, Xinhao Yan, Yan Zeng, Qixuan Zhang, Lan Xu, Wei Yang, Jiayuan Gu, and Jingyi Yu. Cast: Component-aligned 3d scene reconstruction from an rgb image, 2025. 4

[28] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. *arXiv preprint arXiv:2210.02747*, 2022. 4

[29] Tianchang Shen, Jacob Munkberg, Jon Hasselgren, Kangxue Yin, Zian Wang, Wenzheng Chen, Zan Gojcic, Sanja Fidler, Nicholas Sharp, and Jun Gao. Flexible isosurface extraction for gradient-based mesh optimization. *ACM Transactions on Graphics (TOG)*, 42(4):1–16, 2023. 4

[30] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, 42(4):139–1, 2023. 4

[31] Yousset I Abdel-Aziz, Hauck Michael Karara, and Michael Hauck. Direct linear transformation from comparator coordinates into object space coordinates in close-range photogrammetry. *Photogrammetric engineering & remote sensing*, 81(2):103–107, 2015. 4

[32] Jiale Xu, Shenghua Gao, and Ying Shan. Freesplatter: Pose-free gaussian splatting for sparse-view 3d reconstruction. *arXiv preprint arXiv:2412.09573*, 2024. 6, 8, 9

[33] Hao He, Yixun Liang, Luozhou Wang, Yuanhao Cai, Xinli Xu, Haoxiang Guo, Xiang Wen, and Yingcong Chen. Lucidfusion: Reconstructing 3d gaussians with arbitrary unposed images. In *Computer Graphics Forum*, volume 44, page e70227. Wiley Online Library, 2025. 6, 8

[34] Mukul Khanna, Yongsen Mao, Hanxiao Jiang, Sanjay Haresh, Brennan Shacklett, Dhruv Batra, Alexander Clegg, Eric Undersander, Angel X Chang, and Manolis Savva. Habitat synthetic scenes dataset (hssd-200): An analysis of 3d scene scale and realism tradeoffs for objectgoal navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16384–16393, 2024. 7

[35] Huan Fu, Rongfei Jia, Lin Gao, Mingming Gong, Binqiang Zhao, Steve Maybank, and Dacheng Tao. 3d-future: 3d furniture shape with texture. *International Journal of Computer Vision*, 129(12):3313–3337, 2021. 7

[36] Jasmine Collins, Shubham Goel, Kenan Deng, Achleshwar Luthra, Leon Xu, Erhan Gundogdu, Xi Zhang, Tomas F Yago Vicente, Thomas Dideriksen, Himanshu Arora, et al. Abo: Dataset and benchmarks for real-world 3d object understanding. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 21126–21136, 2022. 7

[37] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 7

[38] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on pattern analysis and machine intelligence*, 13(4):376–380, 2002. 13

[39] Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, et al. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first international conference on machine learning*, 2024. 16

# A Technical appendices and supplementary material

## A.1 Alignment protocol

Our baselines differ in what they natively output (input-view pointmaps, canonical pose-free 3D, or input-camera poses), so a single shared alignment would unfairly discard the structure each method already provides. We therefore align each prediction with the protocol best matched to its output family before evaluating, and **GenSfM** is evaluated under each applicable family using that family's protocol so all comparisons within a table are like-for-like.

**Input-view pointmap least squares (Table 2).** This protocol applies to feed-forward reconstruction methods (LucidFusion, FreeSplatter, ReconViaGen), which already place geometry in a noisy estimate of the input-camera frame. For each input view, we back-project the per-pixel predicted depth at the predicted camera into 3D, alpha-mask the result, randomly subsample to at most 50K points per view, and pair it with the GT pointmap obtained by back-projecting GT depth at the GT camera. Stacking the per-view pairs with equal aggregate weight per view yields one set of 3D-to-3D correspondences per object, which we fit with a closed-form Umeyama similarity [38]. The resulting transform is applied once to the full predicted 3D before all photometric and geometric evaluation; we log the per-object LSQ residual as an alignment-fidelity diagnostic. Note that this protocol exercises both halves of feed-forward reconstruction: the matched correspondences require the predicted geometry to align with the predicted cameras, so a method with weak geometry, weak pose recovery, or any mismatch between the two will produce noisy correspondences and a loose Umeyama fit, inflating all downstream metrics.

**Similarity ICP (Table 3).** This protocol applies to 3D generation methods (TRELLIS-S, TRELLIS-M, SAM 3D), which output a canonical pose-free 3D, and to **GenSfM** when evaluated under the same family. To obtain the prediction-side point cloud uniformly across output formats (whether the prediction is a mesh or a set of 3D Gaussians), we render depth from a set of orbiting random views around the prediction and back-project the depth maps into 3D, which yields a well-covered surface point cloud without surface-sampling biases specific to the underlying representation. The GT-side cloud is sampled directly on the GT mesh surface. We initialize alignment by sweeping the 24 orientation-preserving signed axis permutations of the PCA-aligned bounding box and keep the candidate with the lowest post-ICP residual; we then refine with similarity ICP and use the resulting transform to map the prediction into the GT frame for evaluation.

## A.2 Pose metric protocol

**Pose metric definitions.** Different pose-predicting baselines fix their world frame in different ways (some anchor to the first input view, some to an internal canonical frame, others through their own internal similarity fit), so absolute predicted extrinsics cannot be directly compared to GT extrinsics. To bypass this gauge ambiguity entirely, all metrics in Table 4 are computed on *unordered pairs* of input views: for each pair, the relative pose between the two views cancels any global rotation, translation, and scale that affects all views jointly, leaving only the camera geometry intrinsic to the pair. We adopt this pairwise form uniformly for both rotation and translation rather than anchoring to one reference view, which would entangle every error with the chosen anchor. For each object, we average each metric over the $N(N-1)/2$ unordered pairs and then average across objects. **Relative Rotation Error (RRE)** is the geodesic angle (in degrees) between the predicted and GT relative rotations of the pair; **Acc.**@$5^\circ$ **/ Acc.**@$15^\circ$ is the fraction of pairs whose RRE

Table 6: **Pose VAE autoencoding accuracy on Toys4k across latent spatial resolutions.** We sweep the latent grid in $\{4^3, 8^3, 16^3\}$ at a fixed 8 channels per token (yielding $\{64,\ 512,\ 4096\}$ latent tokens respectively), encode each held-out $64^3$ ground-truth UV volume, decode, and recover camera pose via the DLT solve. Each cell reports *median / mean / 95% quantile* over $\sim$77K (image, pose) pairs. UV distance is in normalized image coordinates ($\times 10^{-4}$); pose errors are in degrees.

| Latent | UV Dist.↓ ($\times 10^{-4}$) | RRE↓ | RTE↓ | RFOV↓ |
|---|---|---|---|---|
| $4^3$ | 4.61 / 5.48 / 10.12 | 0.217 / 0.388 / 0.987 | 0.209 / 0.380 / 0.986 | 0.077 / 0.136 / 0.264 |
| $8^3$ | 7.06 / 7.85 / 12.41 | 0.202 / 0.373 / 0.919 | 0.200 / 0.366 / 0.924 | 0.063 / 0.121 / 0.234 |
| $16^3$ | 4.58 / 4.88 / 6.78 | 0.241 / 0.402 / 1.031 | 0.235 / 0.393 / 1.028 | 0.090 / 0.147 / 0.282 |

falls below the corresponding threshold; **Relative Translation Error (RTE)** is the angular error (in degrees) between the predicted and GT relative-translation directions, an angular form chosen because absolute translation magnitude is itself gauge-dependent (a global similarity scale can be reabsorbed by any consistent rescaling of all camera centers); and **Field-of-View Error (RFOV)** is the mean absolute difference (in degrees) between predicted and GT FOV, computed directly on intrinsics without any pose alignment, since intrinsics are independent of the world frame. To score baselines with a coarse front/back chirality ambiguity on geometric correctness rather than chirality alone, all angular metrics (RRE, Acc.@$\{5^\circ, 15^\circ\}$, RTE) are read off after collapsing $R_{\text{pred}} R_{\text{gt}}^\top$ under the $\mathrm{diag}(\pm 1, \pm 1, -1)$ sign-flip equivalence class. The same rule is applied uniformly to all methods, including ours, to compare geometric recovery under a shared front/back equivalence. Chirality remains relevant for camera-dependent applications, so we explicitly note that this metric does not assess that failure mode.

### A.3 Pose VAE

**Architecture and training.** The pose VAE is a 3D-convolutional encoder–decoder over a $64^3$ voxel grid with a deliberately aggressive bottleneck: four $2\times$ spatial downsamples compress the input into a $4^3$ token grid with 8 channels per token. This compression is feasible because each per-voxel target is a smooth function of voxel position and a low-DoF camera, so the latent does not need to absorb high-frequency content. Per voxel the encoder receives three input channels (a binary in-frustum indicator plus the two pixel coordinates obtained by projecting the voxel center under the input camera), and the decoder emits the same three channels through a classification head (visibility mask) and a regression head (pixel UV).

We use a dice loss on the visibility classifier and an L1 loss on the UV regression, with a small KL weight ($10^{-3}$) so the latent stays close to a standard Gaussian for the downstream diffusion stage. The training distribution matches inference: each minibatch sample takes a real camera pose from the multi-view renders of a sampled training object and projects it on the fly into the $64^3$ voxel grid. We optimize for 1M steps with AdamW (lr $3\times 10^{-4}$, weight decay $10^{-3}$) on a cosine schedule.

**Out-of-frame voxels.** Voxel centers can project outside the input image. We handle this by clamping target UVs to $[0, 1]^2$ rather than dropping out-of-frame voxels, keeping the decoder output well-defined everywhere. The visibility channel encodes the same in-/out-of-frame information as a binary indicator with a small $1\%$ inset margin ($0.01 \le u, v \le 0.99$); the inference DLT solve uses this channel to restrict to in-frustum voxels, so saturated boundary UVs never contaminate the recovered pose. The UV reconstruction loss is applied uniformly to all voxels rather than only the in-frustum subset, keeping saturated-boundary regions consistent between encoder and decoder.

**Pose autoencoding accuracy.** Table 6 reports the autoencoding fidelity of the trained pose VAE on the Toys4k held-out split, sweeping the latent spatial resolution. We encode each held-out $64^3$ ground-truth UV volume to the chosen latent grid, decode back, and recover camera pose via a DLT solve over in-frustum voxels. **UV Dist.** is the per-voxel L2 distance in normalized image coordinates; **RRE**, **RTE**, and **RFOV** are the standard rotation, translation-direction, and FOV errors in degrees. All three latent resolutions deliver near-identical pose accuracy (within a few hundredths of a degree on RRE/RTE/RFOV), and even the worst row stays an order of magnitude below the downstream pose-prediction residuals in Table 4, so the latent is not the bottleneck on overall pose accuracy. We adopt the $4^3$ latent in the main paper as the most compute-efficient operating point, since the larger grids carry $8\times$ and $64\times$ more tokens at no measurable pose gain.

### A.4 Image conditioning across stages

**Coarse stage: dual-stream encoder and token fusion.** Each view is encoded by two complementary streams that fuse into a single per-view token sequence for the coarse DiT to cross-attend to. A frozen DINOv2-L/14 with registers at $518\times518$ produces $1374$ tokens per view (CLS + 4 registers + a $37\times37$ patch grid) at $1024$ channels, supplying object-level semantics. In parallel, a lightweight 4-level CNN at $296\times296$ ($5\times5$ convolutions, strides $\{1, 2, 2, 2\}$, output channels $[6, 6, 6, 8]$) is patchified down to the same $37\times37$ grid, with per-token contributions $[384, 96, 24, 8]$ channel-concatenating into a $512$-dim local-texture feature. We append the $512$ local channels onto the $1024$ semantic channels at every patch token (non-patch tokens zero-pad on the local side), giving a uniform per-token width of $1536$ and a cross-attention key/value of length $N \times 1374$. The two streams are deliberately complementary: semantic anchors pose to image content, while local texture restores the pixel-faithful appearance cues the semantic encoder has discarded, letting cross-attention recover accurate cameras without polluting the joint self-attention with raw image tokens.

**Fine stage: visual pyramid and per-voxel projection.** The fine stage refines up to $L \leq 32{,}768$ active voxels at $64^3$ under the cameras already recovered by the coarse stage. Cross-attending every voxel to thousands of image tokens per view is intractable, so we replace cross-attention with per-voxel pixel-aligned feature sampling: each voxel projects to $\mathbf{u}_{j,i}$ in view $i$ under the predicted camera, and visual features are bilinearly sampled at that pixel. The visual encoder is a ConvNeXt-style pyramid run once per view at $256\times256$: four stages, each one ConvNeXt block with a $7\times7$ depthwise kernel at internal width 32, with a $2\times$ downsample between stages. Each stage exposes a $4$-channel tap (LayerNorm + $1\times1$ Conv); the four taps sampled at $\mathbf{u}_{j,i}$ and channel-concatenated yield a $16$-channel multi-scale texture feature $\mathbf{c}_j^{\text{tex}}$. We pair it with an $8$-channel semantic feature $\mathbf{c}_j^{\text{sem}}$: the coarse-stage DINOv2 patch tokens (reused, no extra DINOv2 pass) are bilinearly sampled at the same per-voxel projection and re-encoded through the frozen SLat encoder, placing them on the SLat target's latent manifold and giving a compact $24$-channel per-voxel conditioning vector. This keeps fine-stage cost proportional to active voxels rather than view tokens, while the multi-scale taps let each voxel combine coarse contour cues with fine local texture without inflating the per-voxel width.

### A.5 Implementation details

**Pose VAE training.** The pose VAE of App. A.3 is trained for 1M steps on $8$ H200 GPUs with batch size $4$ per GPU, using AdamW (lr $3\times10^{-4}$, weight decay $10^{-3}$) under a cosine schedule with $1000$ warmup steps decaying to $10^{-5}$, EMA $0.9999$, and FP16 mixed precision. The same recipe trains the $4^3$, $8^3$, and $16^3$ latent variants of Tab. 6.

**Coarse-stage shape-pose DiT training.** The coarse stage is fine-tuned for 1M steps on $8$ H200 GPUs at global batch size $64$ ($8$ per GPU, split into two gradient-accumulation microbatches), initialized from a prior single-view coarse-stage checkpoint of ours (itself from the pretrained TRELLIS-image-large SS DiT). Each instance contributes $N{=}4$ random condition views uniformly sampled from the $24$ rendered views of the object; each input image is crop-resized at a $1.2\times$ tightness ratio around the alpha-mask bounding box and bilinearly resampled to $518\times518$ for the frozen DINOv2 ViT-L/14-with-registers encoder. The DiT operates on $4096$ shape, $256$ pose, and $128$ shared register tokens, with cross-attention over $N \times 1374 = 5496$ image-condition tokens augmented by the convolutional pyramid of Eq. 6. We use AdamW (lr $10^{-4}$, no weight decay), CFG dropout $0.1$, the same cosine schedule and EMA as above, and timesteps drawn from a logit-normal distribution with mean $1.0$ and std $1.0$ ($\sigma_{\min}{=}10^{-5}$). At each step we randomly choose between two timestep-coupling modes: with probability $0.8$, the shape stream and all pose streams share a single timestep; with probability $0.2$, each stream draws an independent timestep. The shared mode dominates so the model stays well-calibrated to the synchronized inference schedule, while the per-pose mode supplies enough timestep diversity that the per-modality AdaLN modulation actually learns to handle independent noise levels.

**Fine-stage SLat DiT training.** The fine stage is fine-tuned independently for 1M steps on $4$ H200 GPUs at global batch size $32$ ($8$ per GPU), initialized from TRELLIS's pretrained SLat DiT checkpoint. It denoises up to $L{\leq}32{,}768$ active voxels at $64^3$; ground-truth cameras and occupancy are used during training so the fine stage learns from clean projections. Each instance again contributes

$N{=}4$ random condition views, with per-voxel features pooled across views by mean (16-channel ConvNeXt-pyramid texture and 8-channel SLat-encoded semantic) and the cross-attention condition the per-view mean of DINOv2 patches. We use AdamW (lr $10^{-4}$, weight decay $10^{-3}$), CFG dropout $0.1$, and the same cosine schedule and logit-normal timestep distribution as the coarse stage.

**Inference.** The coarse stage runs a 25-step flow-Euler sampler over the joint shape-pose sequence at classifier-free guidance strength $7.5$, active on the noise interval $t \in [0.5, 1.0]$ with a guidance time-rescale factor of $3.0$ [39]. We then decode each per-view pose latent through the pose VAE and recover $\mathbf{P}_i^*$ by the DLT solve of Eq. 2. The fine stage runs a 25-step flow-Euler sampler with sequential multi-view fusion at CFG strength $3.0$ on the same $[0.5, 1.0]$ interval and rescale factor, conditioned on the recovered cameras and on the per-voxel pose-sampled features of Eq. 7.

### A.6 Broader impacts

**GENSFM** produces a 3D representation of an object from a few uncalibrated images, jointly recovering shape and camera. The most direct positive applications are in robotics (object-level scene understanding from a small number of viewpoints), augmented and virtual reality (capturing personal items as 3D assets for placement in virtual environments), and content creation (3D meshes from reference photos as a starting point for downstream editing). Joint recovery of camera intrinsics and extrinsics with shape removes the calibration friction that has limited deployment of feed-forward reconstruction methods in casually-captured settings.

The negative considerations are those shared with image-conditioned 3D generation more broadly: the resulting 3D models could be used to fabricate convincing 3D content of real-world objects, including objects belonging to others, without their consent; coupled with downstream rendering pipelines, this could enable image-only digital reproduction at a scale that traditional photogrammetry has not previously allowed. The model is trained only on common, non-sensitive object categories from public 3D-asset datasets, and recovery quality degrades on out-of-distribution categories, which provides a partial natural mitigation. Released model checkpoints will carry an attribution requirement and an acceptable-use note as the safeguard accompanying our open release.

## NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and Sec. 1 state our contributions: recasting a native 3D diffusion model as the primary inference mechanism for sparse-view object reconstruction, a 3D-native multi-view pose tokenization in a unified multi-modal sequence, and a pose-aware fine stage; the experimental results in Sec. 5 on Toys4k and GSO match these claims.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Limitations and future work are discussed in the Conclusion (Sec. 6): experiments are restricted to object-centric inputs at $N{=}4$, and the model inherits the object distribution and resolution limits of its TRELLIS backbone.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include theoretical results requiring formal proofs.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Architecture, training schedule, datasets, alignment protocols, pose-metric definitions, and component-level architectural specifications are documented in Sec. 5 and the supplementary appendices (App. A.1, App. A.2, App. A.3, App. A.4).

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Source code, installation and data-preparation instructions, and inference checkpoints are publicly available at `https://github.com/dotb1ue/GenSfm` and `https://huggingface.co/dotblue/GenSfm`. The public release does not package all artifacts required for exact reproduction of every reported experiment; for example, the inference bundle omits the UV encoder needed to prepare training targets. We therefore retain [No] for this full-reproduction requirement.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Sec. 5.1 and App. A.5 list datasets, splits, optimizer (AdamW), learning rates, batch sizes, training steps, and sampler settings; App. A.3 and App. A.4 provide architectural specifications.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: We report metrics aggregated over 800 object–viewset instances from 400 unique held-out objects per dataset. The current paper does not report confidence intervals or variability across independent training seeds.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Pose VAE, SS DiT, and SLat DiT train for 1M steps on $8/8/4$ H200 GPUs with global batches $32/64/32$, respectively; see the supplementary implementation details.

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The work uses publicly released 3D-asset datasets (Objaverse-XL, HSSD, 3D-FUTURE, ABO, Toys4k, GSO) and pretrained models, and conforms with the NeurIPS Code of Ethics.

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Both positive and negative societal impacts are discussed in App. A.6: enabling sparse-view 3D capture for robotics, AR, and content creation; risks of unauthorized 3D reproduction of real-world objects.

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The model performs object-centric 3D reconstruction from images and does not pose high-risk dual-use concerns beyond the standard generative-modeling considerations noted in App. A.6.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All used datasets (Objaverse-XL, HSSD, 3D-FUTURE, ABO, Toys4k, GSO) and pretrained models (TRELLIS, CUPID, DINOv2) are publicly available under their respective licenses and are properly cited; baselines (LucidFusion, FreeSplatter, ReconViaGen, VGGT, SAM 3D, TRELLIS-S/M) are cited and used per their public release terms.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The released code and inference checkpoints are documented at https://github.com/dotb1ue/GenSfm and https://huggingface.co/dotblue/GenSfm, including installation, input preparation, inference usage, and licenses. Architecture, training, and evaluation details are provided in Sec. 3–5 and the supplementary appendices.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper does not involve research with human subjects.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: LLMs were not used as an important, original, or non-standard component of the method.